



JFN

Attorney Docket No. 1793.1181

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Sung-hyu HAN et al.

Application No.: 10/763,381

Group Art Unit: 2611

Filed: January 26, 2004

Examiner: Dac V. Ha

For: SOFT DEMODULATION METHOD AND APPARATUS

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN APPLICATION IN ACCORDANCE  
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents  
PO Box 1450  
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicants submit herewith a certified copy of the following foreign application:

Korean Patent Application No. 2003-5305 filed January 27, 2003

It is respectfully requested that the applicants be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STEIN, MCEWEN & BUI, LLP

Date: 7/16/07

By: Douglas Rodriguez,  
Douglas X. Rodriguez  
Registration No. 47,269

1400 Eye St., NW  
Suite 300  
Washington, D.C. 20005  
Telephone: (202) 216-9505  
Facsimile: (202) 216-9510



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

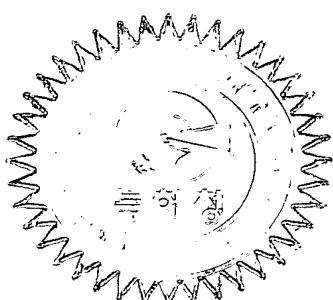
This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원번호 : 10-2003-0005305  
Application Number

출원년월일 : 2003년 01월 27일  
Date of Application JAN 27, 2003

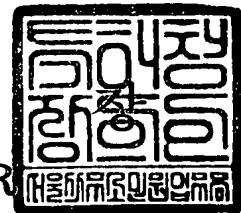
출원인 : 삼성전자주식회사  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.

2004년 01월 15일



특허청

COMMISSIONER



## 【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0013
【제출일자】	2003.01.27
【국제특허분류】	G11B
【발명의 명칭】	소프트 복조 방법 및 소프트 복조 장치
【발명의 영문명칭】	A soft demodulation method and a soft demodulation apparatus
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	2003-003435-0
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2003-003436-7
【발명자】	
【성명의 국문표기】	한성희
【성명의 영문표기】	HAN, Sung Hyu
【주민등록번호】	710223-1063223
【우편번호】	135-220
【주소】	서울특별시 강남구 수서동 신동아아파트 704동 1203호
【국적】	KR
【발명자】	
【성명의 국문표기】	김기현
【성명의 영문표기】	KIM, Ki Hyun
【주민등록번호】	691220-1053119
【우편번호】	463-703

【주소】 경기도 성남시 분당구 구미동(무지개마을) 대림아파트 103동  
1103호

【국적】 KR

【발명자】

【성명의 국문표기】 박인식

【성명의 영문표기】 PARK, In Sik

【주민등록번호】 570925-1093520

【우편번호】 442-470

【주소】 경기도 수원시 팔달구 영통동 신나무실 615동 801호

【국적】 KR

【발명자】

【성명의 국문표기】 이윤우

【성명의 영문표기】 LEE, Yoon Woo

【주민등록번호】 650525-1177713

【우편번호】 442-470

【주소】 경기도 수원시 팔달구 영통동 현대아파트 726동 303호

【국적】 KR

【취지】 특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인  
이영필 (인) 대리인  
이해영 (인)

【수수료】

【기본출원료】	20	면	29,000	원
【가산출원료】	12	면	12,000	원
【우선권주장료】	0	건	0	원
【심사청구료】	0	항	0	원
【합계】	41,000			원

【첨부서류】 1. 요약서·명세서(도면)\_1통

**【요약서】****【요약】**

본 발명에 따라 소프트 복조 방법 및 소프트 복조 장치가 개시된다.

상기 본 발명에 따른 소프트 복조 방법은, a) 채널로부터 수신된 코드 워드를 소정수의 비트 단위마다 부분합을 계산하는 단계와, b) 디코딩 테이블의 각 엔트리값 계산시 상기 각 엔트리를 상기 소정수의 비트 단위로 분할하여 상기 분할된 소정수의 비트 단위에 대응하는 상기 부분합을 참조하여 엔트리값들을 계산하는 단계와, c) 상기 엔트리 값들로부터 최대값을 검출하고 LLR 값을 계산하는 단계를 포함한다. 이상과 같은 본 발명에 의하면, 채널의 출력신호로부터 코드의 소프트 복조 수행시 필요한 계산량과 복잡도를 상당히 감소시킬 수 있다.

**【대표도】**

도 6

**【명세서】****【발명의 명칭】**

소프트 복조 방법 및 소프트 복조 장치{A soft demodulation method and a soft demodulation apparatus}

**【도면의 간단한 설명】**

도 1은 종래기술에 따른 데이터 부호화/복호화 장치의 개략적인 블럭도,

도 2는 도 1에 도시된 소프트 복조부의 구체적인 구성을 도시하는 도면,

도 3은 도 2에 도시된 디코딩 테이블의 일 예를 도시한 도면,

도 4는 도 2에 도시된 연산부에서의 연산 과정의 일부를 보여주는 도면,

도 5는 본 발명에 따른 소프트 복조 과정의 제1실시예를 설명하는 흐름도,

도 6은 도 5에 도시된 제1실시예에 따른 소프트 복조 장치 구성의 일 예를 나타내는 블럭도,

도 7은 도 6에 도시된 기준 엔트리를 설명하기 위한 도면,

도 8은 도 6에 도시된 부분합 테이블의 일 예를 도시한 도면,

도 9는 도 6에 도시된 부분합 테이블의 다른 예를 도시한 도면,

도 10은 도 6에 도시된 부분합 테이블이 또다른 예를 도시한 도면,

도 11은 본 발명에 따른 소프트 복조 과정의 제2실시예를 설명하는 흐름도,

도 12는 도 11에 도시된 제2실시예에 따른 소프트 복조 장치 구성의 일 예를 나타내는 블럭도.

**【발명의 상세한 설명】****【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <13> 본 발명은 데이터 복호화에 관한 것으로, 좀더 구체적으로는 소프트 복조 방법 및 소프트 복조 장치에 관한 것이다.
- <14> 일반적으로 데이터의 심볼간 간섭 ISI(Inter Symbol Interference)이 존재하는 환경, 예컨대 고밀도 광 기록매체에서 ISI에 의한 문제를 고려하여 런길이 제한(run-length limited: RLL) 코드를 사용하여 데이터를 변조한다.
- <15> 종래의 RLL 코드를 복호화하는 경우를 보면 채널 복조기, 예컨대 비터비 복호기에서 채널을 통해 입력된 신호로부터 코드워드들을 검출하고, RLL 복호기에서 디코딩 테이블을 이용하여 코드워드들을 데이터 워드들로 복호화한다.
- <16> 최근에는 소프트 복조기 및 소프트 복호기가 사용되고 있다. 상술한 비터비 복호기의 출력은 에러를 포함하는 코드워드 자체가 되어 "1" 또는 "-1"의 값만 가진다. 이러한 방식을 하드 복조라고 할 때 이에 대응하여 소프트 복조 방식에서는 소프트 복조기가 채널 신호를 입력 받아 코드워드의 확률값을 출력한다. 즉, "1" 또는 "-1"이 아닌 "0.8", "-0.8" 등의 아날로그적인 값을 가지며 그 자체에 코드값이 "1" 또는 "-1"이 될 확률이 내포된다. 소프트 복조기는 이러한 코드워드의 확률값을 나타내는 데이터를 입력받아 데이터 워드의 확률값을 출력하고, 터보 복호기와 같은 소프트 복호기에서 데이터 워드의 확률값을 입력받아 최종적으로 데이터 워드로 복호화한다.

- <17> 소프트 변조를 필요로 하는 터보 디코딩에 관한 참조 문헌으로는 Laura L. McPheters and Steven W. McLaughlin, "Turbo-Coded Optical Recording Channels with DVD Minimum Mark Size", IEEE Transactions on Magnetics, vol.38, no. 1, January pp 298-302이 있다.
- <18> 도 1에 종래기술에 따른 부호화/복호화 장치가 도시되어 있다.
- <19> 상기 부호화/복호화 장치(100)는 터보/LDPC 부호화부(110)와, 변조부(120)와, 기록/독출부(130)와, 소프트 복조부(150)와, 터보/LDPC 복호화부(160)를 포함한다.
- <20> 터보/LDPC 부호화부(110)는 입력되는 데이터를 여러 정정을 수행하기 위한 소정의 부호화 방식, 예를 들어, LDPC(Low Density Parity Check Code) 부호화 방식이나 터보 부호화 방식과 같은 소프트 부호화 방식을 이용하여 부호화한다.
- <21> 변조부(120)는 상기 터보/LDPC 부호화(110)로부터 출력된 데이터를 소정의 방법, 예를 들어, RLL 코드 등을 이용하여 변조한다.
- <22> 기록/독출부(130)는 이와 같이 변조된 데이터를 기록 매체에 기록하고, 기록 매체에 기록된 데이터를 독출한다.
- <23> 소프트 복조부(150)는 코드워드의 확률값을 나타내는 데이터를 기록/독출부(130)로부터 입력받아 데이터 워드를 구성하는 각각의 비트의 확률을 나타내는 값인 LLR(Log Likelihood Ratio)을 출력한다.
- <24> 터보/LDPC 복호화부(160)는 소프트 복조부(150)로부터 출력된 데이터를 수신하여 소정의 부호화 방식에 대응하여 소프트 복호화를 수행하고 복호화된 데이터를 출력한다.
- <25> 상기 소프트 복조부(150)는 LLR 값을 구하기 위해 APP(A Posteriori Probability)( $d_k=1$ )과 APP( $d_k=0$ )을 각각 계산한다.

<26> APP( $d_k=1$ )은 복조된 데이터가 "1"일 확률을 나타내는 값이며, APP( $d_k=0$ )은 복조된 데이터가 "0"일 확률을 나타내는 값으로, 이 값은 다음과 같은 방법으로 구한다. 만일 데이터 워드를 결정하기 위한 코드워드의 길이가 N 비트 라면, 데이터 워드의 1 비트가 1이 되도록 하는 코드워드에 대해 각 비트별로  $(r_m - (2*c_m - 1))^2$ 의 값을 구하여 더한다. 여기서  $m=1, \dots, N$ 의 값이다. 이 값을 상술한 데이터 워드의 1 비트가 1이 되도록 하는 모든 코드워드 M개에 대해 지수(exponential)값을 취하여 더하면 APP( $d_k=1$ )의 값이 구해진다. 이를 식으로 나타내면 아래와 같다.

$$<27> \text{【수학식 1】 } APP(d_k=1) = \sum_{j \in S_1(k)} \exp [r_m^j - (2 * c_m^j - 1)^2]$$

<28> 여기서, j는 j번째 데이터 워드가 1이 된다는 의미이다.

<29> APP( $d_k=0$ )에 대해서도 마찬가지 방법으로 구하는 데 즉, 아래 수학식 2에 도시된 바와 같다.

$$<30> \text{【수학식 2】 } APP(d_k=0) = \sum_{j \in S_0(k)} \exp [r_m^j - (2 * c_m^j - 1)^2]$$

<31> LLR( $d_k$ )는 수신된 코드워드에서 복조될 데이터 워드  $d_k$ 의 1비트가 0인 확률과 1인 확률의 비에 대해 지수를 취한 값이므로 다음과 같이 나타낼 수 있으며, 이 값을 소프트 복조부의 출력으로 사용한다.

$$<32> \text{【수학식 3】 } LLR(d_k) = \log \frac{(APP(d_k=1))}{(APP(d_k=0))}$$

<33> 그러나, 지수 계산이 복잡할 때는 APP 계산시 지수 계산을 하지 않고 최대값을 취하여 LLR을 계산하는 방법이 있다. 즉, 수학식 3에서 표현된 계산 연산 대신  $APP(d_k=1) - APP(d_k=0)$

의 값을 계산해서 LLR( $d_k$ )를 구한다. 이와 같이 최대값 계산에 의한 소프트 복조부(150)의 구체적인 구성이 도 2에 도시되어 있다.

<34> 상기 소프트 복조부(150)는 디코딩 테이블(151)과, 연산부(152)와, 최대값 검출부(153)와, LLR 계산부(154)를 포함한다.

<35> 디코딩 테이블(151)은 도 3에 도시된 바와 같다.

<36> 연산부(152)는 도 4에 도시된 방법에 의해 디코딩 테이블(151)로부터 출력되는 테이블 엔트리를 이용하여 계산을 한다. 수신된 코드신호(410)의 각 데이터와 테이블 엔트리(420)의 각 비트를 대응시켜, 대응되는 테이블 엔트리의 비트가 0이면 수신된 코드신호에 -1을 곱하고, 대응되는 테이블 엔트리의 비트가 1이면 +1을 곱하여 모두 합한 계산 결과(430)를 출력한다.

<37> 예를 들어, 도 4에서 채널로부터 수신된 코드신호(410)가  $r_0, r_1, \dots, r_{17}$ 이고, 디코딩 테이블(151)로부터의 테이블 엔트리가 "000101...100"인 경우, 테이블 엔트리의 4번째, 6번째, 9번째, 11번째, 14번째, 16번째 비트가 1이고, 나머지 비트들은 모두 0이다. 그러므로, 수신된 코드신호  $r_3, r_5, r_8, r_{10}, r_{13}, r_{15}$ 에는 각각 +1을 곱하고, 그 외 나머지 코드신호에 대해서는 각각 -1을 곱하여 더하면 도 4에 도시된 바와 같은 계산결과(430)가 나온다. 연산부는 이와 같은 계산을 디코딩 테이블에 속한 테이블 엔트리의 수만큼 행한다.

<38> 최대값 검출부(153)는 연산부(152)로부터의 테이블 엔트리의 수만큼의 계산결과를 수신하여 그 값들중에서 최대값을 검출한다.

<39> LLR 계산부(154)는 다음 식과 같이 최대값 검출부(153)로부터 전송되는 0에 대한 최대값과 1에 대한 최대값을 수신하여 1에 대한 최대값에서 0에 대한 최대값을 빼는 연산을 수행하여 LLR을 계산하고 이러한 LLR 값을 출력한다.

&lt;40&gt;

$$\text{【수학식 4】 } LLR(d_k) = \max_{j \in S_1(k)} \left[ \sum_{m=0}^{k-1} r_m^j * 2(c_m^j - 1) \right] - \max_{j \in S_0(k)} \left[ \sum_{m=0}^{k-1} r_m^j * 2(c_m^j - 1) \right]$$

&lt;41&gt;

또한,  $LLR(d_{k+1})$ 은 수학식4에 표시된  $LLR(d_k)$ 에서  $k$  대신  $k+1$ 을 대입하면 된다. 즉,

&lt;42&gt;

$$\text{【수학식 5】 } LLR(d_{k+1}) = \max_{j \in S_1(k+1)} \left[ \sum_{m=0}^{k-1} r_m^j * 2(c_m^j - 1) \right] - \max_{j \in S_0(k+1)} \left[ \sum_{m=0}^{k-1} r_m^j * 2(c_m^j - 1) \right]$$

&lt;43&gt;

여기서,  $S_0(k+1)$ 은 도 3에 도시된 바와 같은 디코딩 테이블의 엔트리들의 집합이고,  $S_1(k+1)$ 은 도시하지 않았지만 도 3에 도시된 바와 유사한 디코딩 테이블의 엔트리들의 집합이다.

&lt;44&gt;

도 1에 도시된 바와 같은 테이블을 참조하여 계산하면, 덧셈 또는 뺄셈이 17(각 테이블 엔트리에 속한 비트에 관한 덧셈 또는 뺄셈의 수)\*5000(테이블 엔트리의 수) = 85000 번이 필요하다.

&lt;45&gt;

상술한 바와 같이, 종래에는 소프트 복조기에 사용되는 APP 디코딩 테이블의 사이즈가 커지면 LLR을 계산하는데 계산 시간이 늘어나고, 디코딩 테이블의 복잡도가 크게 증가하는 문제점이 있다.

### 【발명이 이루고자 하는 기술적 과제】

&lt;46&gt;

본 발명은 상기와 같은 과제를 해결하여 소프트 복조 수행시 디코딩 테이블의 복잡도 및 계산량을 감소시킬 수 있는 소프트 복조 방법 및 소프트 복조 장치를 제공하는 것을 목적으로 한다.

### 【발명의 구성 및 작용】

&lt;47&gt;

상기와 같은 과제를 해결하기 위해 본 발명이 착안한 것은 하는 것이다.

<48> 상기와 같은 과제를 해결하기 위한 본 발명의 하나의 특징은, 소프트 복조 방법에 있어서, a) 채널로부터 수신된 코드 워드를 소정수의 비트 단위마다 부분합을 계산하는 단계와, b) 디코딩 테이블의 각 엔트리값 계산시 상기 각 엔트리를 상기 소정수의 비트 단위로 분할하여 상기 분할된 소정수의 비트 단위에 대응하는 상기 부분합을 참조하여 엔트리값들을 계산하는 단계와, c) 상기 엔트리 값들로부터 최대값을 검출하고 LLR 값을 계산하는 단계를 포함하는 것이다.

<49> 여기서, 바람직하게는, 상기 a) 단계는, 디코딩 테이블의 엔트리에서 소정수의 비트단위마다 나타날 수 있는 비트 조합으로 이루어진 기준 엔트리를 참조하여 상기 채널로부터 수신된 코드워드를 소정수의 비트 단위마다 부분합을 계산한다.

<50> 또한, 바람직하게는, 상기 b) 단계는, b1) 디코딩 테이블의 각 엔트리를 상기 소정수의 비트 단위로 분할하는 단계와, b2) 상기 분할된 소정수의 비트 단위마다 그에 대응하는 부분합을 가져와서 단위당 부분합을 모두 더하는 단계를 포함한다.

<51> 또한, 바람직하게는, 상기 b) 단계는, 디코딩 테이블의 엔트리값 계산시, 상기 대응하는 부분합이 소정의 임계치를 초과하지 않는 경우에 그 엔트리값은 계산하지 않는다.

<52> 본 발명의 다른 특징은, 소프트 복조 장치에 있어서, 채널로부터 수신된 코드 워드를 소정수의 비트 단위마다 부분합을 계산하는 부분합 계산부와, 디코딩 테이블의 각 엔트리값 계산시 상기 각 엔트리를 상기 소정수의 비트 단위로 분할하여 상기 분할된 소정수의 비트 단위에 대응하는 상기 부분합을 참조하여 엔트리값들을 계산하는 엔트리 계산부와, 상기 엔트리 값들로부터 최대값을 검출하는 최대값 검출부와, 상기 검출된 최대값을 이용하여 LLR 값을 계산하는 LLR 계산부를 포함한다.

- <53> 이제, 첨부된 도면을 참조하여 본 발명을 상세히 설명한다.
- <54> 도 5는 본 발명에 따른 소프트 복조 과정의 제1실시예를 설명하는 흐름도이다.
- <55> 먼저, 채널로부터 코드 신호를 수신한다.
- <56> 다음, 기준 엔트리를 참조하여 상기 수신된 코드신호에 대한 부분합을 계산한다.
- <57> 기준 엔트리라는 것은 디코딩 테이블의 각 엔트리에 포함된 비트들중 소정 개수의 비트로 이루어질 수 있는 가지 수를 나타내는 단위 엔트리를 말한다. 디코딩 테이블의 엔트리는 도 3에 도시된 바와 같은 형태 즉, 각 엔트리는 18개의 비트로 이루어지는 형태를 가지지만, 각 엔트리가  $2^{18}$ 개의 비트 조합을 가지는 것은 아니고, 18개의 비트는 3비트 단위로 쪼개져서, 각 3비트 단위 엔트리는 "000", "001", "010", "100", "101" 의 5가지의 값만을 가질 수 있다. 이하에서는 상기 "000", "001", "010", "100", "101" 을 기준 엔트리라 부르기로 한다.
- <58> 다음, 수신된 코드신호를 3비트 단위로 상기 기준 엔트리를 참조하여 연산을 하여 부분합을 계산한다. 예를 들어, 처음 3비트  $r_0, r_1, r_2$  에 대해서 상기 기준 엔트리 "000", "001", "010", "100", "101"를 적용하여 연산을 하면, "000"에 대한 부분합  $-r_0-r_1-r_2$ , "001"에 대한 부분합  $-r_0-r_1+r_2$ , "010"에 대한 부분합  $-r_0+r_1-r_2$ , "100"에 대한 부분합  $+r_0-r_1-r_2$ , "101"에 대한 부분합  $+r_0-r_1+r_2$  5개가 계산된다. 그리고, 다음 3비트  $r_3, r_4, r_5$  에 대해서 부분합  $-r_3-r_4-r_5$ ,  $-r_3-r_4+r_5$ ,  $-r_3+r_4-r_5$ ,  $+r_3-r_4-r_5$ ,  $+r_3-r_4+r_5$  5개가 계산된다. 이러한 방법으로 수신된 코드신호 18비트에 대해서 각 3비트 단위로 5개의 부분합을 계산한다.
- <59> 다음, 디코딩 테이블의 엔트리를 참조하여 상기 계산된 부분합으로부터 각 엔트리 값들을 계산한다. 즉, 디코딩 테이블의 엔트리가 000 101 001 010 010 100 이면, 상기 엔트리의 3비트 단위 엔트리씩 검색을 하여 그에 대응하여 계산된 부분합을 가져와서 전체 엔트리 합을

계산한다. 다시 말하면, 상기 디코딩 테이블의 엔트리의 처음 3비트가 000 이므로 계산된 부분합  $-r_0-r_1-r_2$  을 가져오고, 다음 3비트 엔트리가 101이므로 계산된 부분합  $+r_3-r_4+r_5$  를 가져오고,... 이런 방식으로 18비트에 대해서 3비트씩 계산된 부분합 6개를 가져와서 모두 더하면 엔트리값이 되는 것이다. 이러한 방식으로 5000개의 엔트리값을 모두 계산한다.

<60> 다음, 계산된 엔트리 값들중에서 최대값을 검출한다.

<61> 그리고나서, LLR값을 계산하고 출력한다.

<62> 이와 같은 방법에 의하면, 예를 들어, 디코딩 테이블에 5000개의 엔트리가 있고, 부분합 테이블을 6개 생성하는 경우, 5(부분합 테이블의 엔트리 개수)\*6(부분합 테이블의 수)\*2(각 엔트리마다 수행되는 연산의 개수)+5000(디코딩 테이블의 엔트리 수)\*5(하나의 엔트리값 계산 시 더해져야 하는 부분합 값의 연산 수)= 25060이 된다.

<63> 도 5에 도시된 제1실시예에 따른 소프트 복조 장치 구성의 일 예가 도 6에 도시되어 있다.

<64> 상기 소프트 복조 장치(600)는 기준 엔트리(610)와, 부분합 계산부(620)와, 부분합 테이블(630)과, 엔트리 계산부(640)와, 최대값 검출부(650)와, LLR 계산부(660)와, 디코딩 테이블(670)을 포함한다.

<65> 기준 엔트리(610)는 도 7에 도시된 바와 같은 엔트리를 가진다. 즉, 기준 엔트리는 5개의 값 "000"(611), "001"(612), "010"(613), "100"(614), "101"(615) 형태를 가진다.

<66> 부분합 계산부(620)는 기준 엔트리(610)로부터 하나의 엔트리씩 그리고 채널로부터 독출된 코드신호를 3비트씩 수신하여, 3비트 단위로 부분합을 계산한다. 즉, 채널로부터 독출된 코드신호가  $r_0$ ,  $r_1$ ,  $r_2$  이고 수신된 기준 엔트리가 "000"이면, 부분합 엔트리  $-r_0 - r_1 - r_2$ 를

계산하고, 다음 기준 엔트리 "001"에 대해 부분합 엔트리는  $-r_0 + r_1 + r_2$  를 계산하고, 다음 기준 엔트리 "010"에 대해서 부분합 엔트리  $-r_0 + r_1 - r_2$  를 계산하고, 다음 기준 엔트리 "100"에 대해서 부분합 엔트리  $+r_0 - r_1 - r_2$  를 계산하고, 다음 기준 엔트리 "101"에 대해서 부분합 엔트리  $+r_0 - r_1 + r_2$  를 계산한다.

<67> 부분합 계산부(620)가 이와 같은 부분합 엔트리 계산을  $r_0, r_1, r_2$  부터  $r_{15}, r_{16}, r_{17}$  까지 수행하면, 예를 들어, 도 8에 도시된 바와 같은 부분합 테이블(800)이 완성된다.

<68> 테이블(800)은 8개의 부분합 테이블(810, 820, 830, 840, 850, 860)을 포함하며, 각 부분합 테이블은 5개의 엔트리를 포함한다.

<69> 엔트리 계산부(640)는 디코딩 테이블(670)의 테이블 엔트리값을 계산하는데, 부분합 테이블(630)에 저장된 계산된 부분합을 이용하여 엔트리값을 계산한다. 즉, 디코딩 테이블(670)의 첫 번째 엔트리가 "000 101 001 010 010 100"이면, 3비트씩 참조하여 그 3비트 엔트리에 해당하는 값을 부분합 테이블(630)에서 검색하고 검색된 값을 가져와서 합연산을 수행한다.

<70> 예를 들어, 도 8에 도시된 부분합 테이블을 참조하면, 상기 첫 번째 엔트리에서 처음 3비트 "000"에 대한 부분합  $-r_0 - r_1 - r_2$ (811)을 부분합 테이블(810)에서 찾고, 다음 3비트 "101"에 대한 부분합  $+r_3 - r_4 + r_5$  (825)을 부분합 테이블(820)에서 찾고, 이런 방법으로 상기 첫 번째 엔트리의 3비트 마다의 부분합을 부분합 테이블에서 찾으면 다음과 같은 값이 될 수 있다.

<71>	디코딩 테이블 엔트리	부분합 테이블 엔트리
	000	$r_0 - r_1 - r_2$
	101	$+ r_3 - r_4 + r_5$
	001	$+ r_6 - r_6 + r_8$
	010	$+ r_9 - r_{10} + r_{11}$
	010	$+ r_{12} - r_{13} + r_{14}$
	100	$+ r_{15} - r_{16} + r_{17}$

<72> 그리고, 상기 찾아진 부분합 테이블 엔트리를 모두 더하면 첫 번째 엔트리에 대한 엔트리값이 된다. 이와 같은 엔트리값 계산을 테이블 엔트리에 포함된 5000개의 엔트리들에 대해서 모두 수행하면 5000 개의 엔트리값들을 얻을 수 있다.

<73> 최대값 검출부(650)는 상기 엔트리 계산부(640)에 의해 구해진 엔트리값들로부터 최대값을 검출한다.

<74> LLR 계산부(660)는 상기 최대값 검출부(650)로부터 출력된 최대값으로부터 LLR을 계산하여 출력한다.

<75> 도 9는 도 6에 도시된 부분합 테이블의 다른 예를 도시한다.

<76> 도 9에 도시된 부분합 테이블(900)은 도 8에 도시된 6개의 부분합 테이블을 각 두 개씩 짹을 지어 하나의 테이블로 만든 것이다. 테이블(810)과 테이블(820)의 엔트리를 조합하여 하나의 테이블(910)을 만들고, 테이블(830)과 테이블(840)의 엔트리를 조합하여 하나의 테이블(920)을 만들고, 테이블(850)과 테이블(860)의 엔트리를 조합하여 하나의 테이블(930)을 만든다. 이와 같이 하여 생성된 각 테이블 (910), (920), (930)은 5개의 엔트리와 5개의 엔트리의 조합인 25개의 엔트리를 가진다.

<77> 이러한 경우의 총 계산량은,

<78>  $30*2(6\text{개의 부분합 테이블을 만드는데 필요한 계산량})$

- <79> + 3\*25(6개의 테이블을 묶어서 3개의 부분합 테이블을 만드는데 필요한 계산량)
- <80> + 5000\*2(최종 엔트리값을 계산하는데 드는 계산량)
- <81> = 10135 가 된다.
- <82> 도 10은 도 6에 도시된 부분합 테이블의 또다른 예를 도시한다.
- <83> 도 10에 도시된 부분합 테이블(1000)은 도 8에 도시된 6개의 부분합 테이블을 각 세 개씩 짹을 지어 하나의 테이블로 만든 것이다. 테이블(810), 테이블(820), 테이블(830)의 엔트리를 조합하여 하나의 테이블(1010)을 만들고, 테이블(840), 테이블(850), 테이블(860)의 엔트리를 조합하여 하나의 테이블(1020)을 만든다. 이와 같이 하여 생성된 테이블 (1010), (1020) 각각은 5\*5\*5의 조합인 125개의 엔트리를 가진다.
- <84> 이러한 경우의 총 계산량은,
- <85> 30\*2(6개의 부분합 테이블을 만드는데 필요한 계산량)
- <86> + 2\*125(6개의 부분합 테이블을 묶어서 2개의 부분합 테이블을 만드는데 필요한 계산량)
- <87> + 5000\*1(최종 엔트리값을 계산하는데 드는 계산량)
- <88> = 5310 이 된다.
- <89> 도 11은 본 발명에 따른 소프트 복조 과정의 제2실시예를 나타낸다.
- <90> 제2실시예에 의한 소프트 복조 방법은 상기 제1실시예에 의한 소프트 복조 수행시에 부분합이 일정값(임계치) 미만이면, 엔트리 계산시에 계산하지 않고 무시함으로써 무용한 계산량의 낭비를 감소시키는 것이다.

<91> 예를 들어, 임계치를 0이라고 설정하였다고 가정하자. 도 3의 첫 번째 엔트리의 원소가 000 101 001 010 010 100 이다. 부분합 테이블이 도 10에 도시된 바와 같이 2블럭으로 이루어져 있다고 가정할 때, 저장된 메모리의 첫 번째 블럭(1010)에서  $r_0 + r_1 + r_2 + r_3 + r_4 + r_5 + r_6 + r_7 + r_8$  와, 두 번째 블럭(1020)에서  $r_9 + r_{10} + r_{11} + r_{12} + r_{13} + r_{14} + r_{15} + r_{16} + r_{17}$  를 더하는데, 만일 첫 번째 또는 두 번째 값이 임계치 0보다 작다면 첫 번째 값과 두 번째 값을 더하는 과정을 무시하고 다음 엔트리 계산으로 넘어간다. 즉, 임계치가 0 이상인 값만 덧셈을 실행하여 맥스 함수의 입력으로 사용한다. 임계치는 채널의 상황에 따라서 달라질 수 있으며, 만일 임계치를 만족하는 값이 없다면 임계치의 값을 낮추어서 다시 계산한다.

<92> 채널로부터 코드 신호를 수신한다.

<93> 다음, 기준 엔트리를 참조하여 부분합을 계산한다. 기준 엔트리를 참조하여 부분합을 계산하는 과정은 제1실시예에서와 동일한 방법에 의한다.

<94> 다음, 엔트리값을 계산할 때, 엔트리값을 모두 더하는 연산을 수행하기 전에 각 부분합이 임계치를 초과하는지 판단한다.

<95> 판단결과, 엔트리값을 계산하기 위한 부분합이 임계치를 초과하지 않는 경우에는 해당 엔트리값을 계산하지 않고 그대로 무시하며, 초과하는 경우에 상기 계산된 부분합을 참조하여 디코딩 테이블의 엔트리값들을 계산한다.

<96> 다음, 계산된 엔트리값들중에서 최대값을 검출한다.

<97> 다음, 검출된 최대값을 이용하여 LLR을 계산한다.

<98> 이와 같이 제2실시예에 의한 소프트 복조 방법은, 제1실시예에 따른 소프트 복조 방법에 의한 계산량 감소외에 임계치를 초과하지 않는 부분합에 대해서는 엔트리 값 계산 자체를 하지

않음으로써, 최대값으로 검출될 확률이 낮은 엔트리들에 대한 무용한 계산량을 더 감소시킬 수 있다.

- <99>        도 12는 도 11에 도시된 제2실시예에 따른 소프트 복조 장치 구성의 일 예를 나타낸다.
- <100>        상기 소프트 복조 장치(1200)는 기준 엔트리(1210)와, 부분합 계산부(1220)와, 부분합 테이블(1230)과, 임계치 판단부(1240)와, 엔트리 계산부(1250)와, 최대값 검출부(1260)와, LLR 계산부(1270)와, 디코딩 테이블(1280)을 포함한다.
- <101>        도 6에 도시된 소프트 복조 장치(600)와 임계치 판단부(1240)의 구성을 제외하고 다른 구성은 모두 동일하다.
- <102>        즉, 기준 엔트리(1210)는 도 7에 도시된 바와 같은 기준 엔트리를 가지며, 부분합 계산부(1220)는 상기 기준 엔트리(1210)를 참조하여 부분합을 계산하고, 부분합 테이블(1230)은 상기 부분합 계산부(1220)에 의해 계산된 부분합을 저장한다.
- <103>        그러면, 엔트리 계산부가 디코딩 테이블(1280)을 참조하여 엔트리값을 계산할 때, 무조건 부분합 테이블(1230)로부터 해당 부분합을 찾아서 엔트리값 계산에 이용하는 것이 아니라, 임계치 판단부(1240)가 먼저 찾아진 부분합을 소정의 임계치와 비교한다.
- <104>        이와 같은 임계치 판단부(1240)의 비교에 의해 임계치를 초과한 부분합에 대해서만 엔트리 계산부(1250)는 엔트리값을 계산하고, 초과하지 않는 부분합에 대해서는 그대로 폐기하여 해당 엔트리 값을 계산하지 않고 다음 엔트리값 계산으로 넘어간다.
- <105>        이와 같이 함으로써 최대값으로 검출될 확률이 낮은 엔트리에 대해서는 그 엔트리값 계산 자체를 수행하지 않음으로써 무용한 계산량의 낭비를 줄일 수 있다.

【발명의 효과】

<106> 이상과 같은 본 발명에 의하면, 채널의 출력신호로부터 RLL 코드의 소프트 복조 수행시 필요한 계산량과 복잡도를 상당히 감소시킬 수 있다.

**【특허청구범위】****【청구항 1】**

소프트 복조 방법에 있어서,

- a) 채널로부터 수신된 코드 워드를 소정수의 비트 단위마다 부분합을 계산하는 단계와,
- b) 디코딩 테이블의 각 엔트리값 계산시 상기 각 엔트리를 상기 소정수의 비트 단위로 분할하여 상기 분할된 소정수의 비트 단위에 대응하는 상기 부분합을 참조하여 엔트리값들을 계산하는 단계와,
- c) 상기 엔트리 값들로부터 최대값을 검출하고 LLR 값을 계산하는 단계를 포함하는 것을 특징으로 하는 소프트 복조 방법.

**【청구항 2】**

제1항에 있어서,

상기 a) 단계는,

디코딩 테이블의 엔트리에서 소정수의 비트단위마다 나타날 수 있는 비트 조합으로 이루 어진 기준 엔트리를 참조하여 상기 채널로부터 수신된 코드워드를 소정수의 비트 단위마다 부 분합을 계산하는 것을 특징으로 하는 소프트 복조 방법.

**【청구항 3】**

제1항에 있어서,

상기 b) 단계는,

- b1) 디코딩 테이블의 각 엔트리를 상기 소정수의 비트 단위로 분할하는 단계와,
- b2) 상기 분할된 소정수의 비트 단위마다 그에 대응하는 부분합을 가져와서 단위당 부분합을 모두 더하는 단계를 포함하는 것을 특징으로 하는 소프트 복조 방법.

#### 【청구항 4】

제3항에 있어서,

상기 소정수의 비트는 3 비트, 6 비트, 9 비트를 포함하는 것을 특징으로 하는 소프트 복조 방법.

#### 【청구항 5】

제2항에 있어서,

상기 기준 엔트리는 000, 001, 010, 100, 101 를 포함하는 것을 특징으로 하는 소프트 복조 방법.

#### 【청구항 6】

제1항에 있어서,

상기 b) 단계는,

디코딩 테이블의 엔트리값 계산시, 상기 대응하는 부분합이 소정의 임계치를 초과하지 않는 경우에 그 엔트리값은 계산하지 않는 것을 특징으로 하는 소프트 복조 방법.

#### 【청구항 7】

소프트 복조 장치에 있어서,

채널로부터 수신된 코드 워드를 소정수의 비트 단위마다 부분합을 계산하는 부분합 계산부와,

디코딩 테이블의 각 엔트리값 계산시 상기 각 엔트리를 상기 소정수의 비트 단위로 분할하여 상기 분할된 소정수의 비트 단위에 대응하는 상기 부분합을 참조하여 엔트리값들을 계산하는 엔트리 계산부와,

상기 엔트리 값들로부터 최대값을 검출하는 최대값 검출부와,  
상기 검출된 최대값을 이용하여 LLR 값을 계산하는 LLR 계산부를 포함하는 것을 특징으로 하는 소프트 복조 장치.

#### 【청구항 8】

제7항에 있어서,  
상기 부분합 계산부는,  
디코딩 테이블의 엔트리에서 소정수의 비트단위마다 나타날 수 있는 비트 조합으로 이루어진 기준 엔트리를 참조하여 상기 채널로부터 수신된 코드워드를 소정수의 비트 단위마다 부분합을 계산하는 것을 특징으로 하는 소프트 복조 장치.

#### 【청구항 9】

제7항에 있어서,  
상기 엔트리 계산부는,  
디코딩 테이블의 각 엔트리를 상기 소정수의 비트 단위로 분할하고, 상기 분할된 소정수의 비트 단위마다 그에 대응하는 부분합을 가져와서 단위당 부분합을 모두 더하는 연산을 수행하는 것을 특징으로 하는 소프트 복조 장치.

#### 【청구항 10】

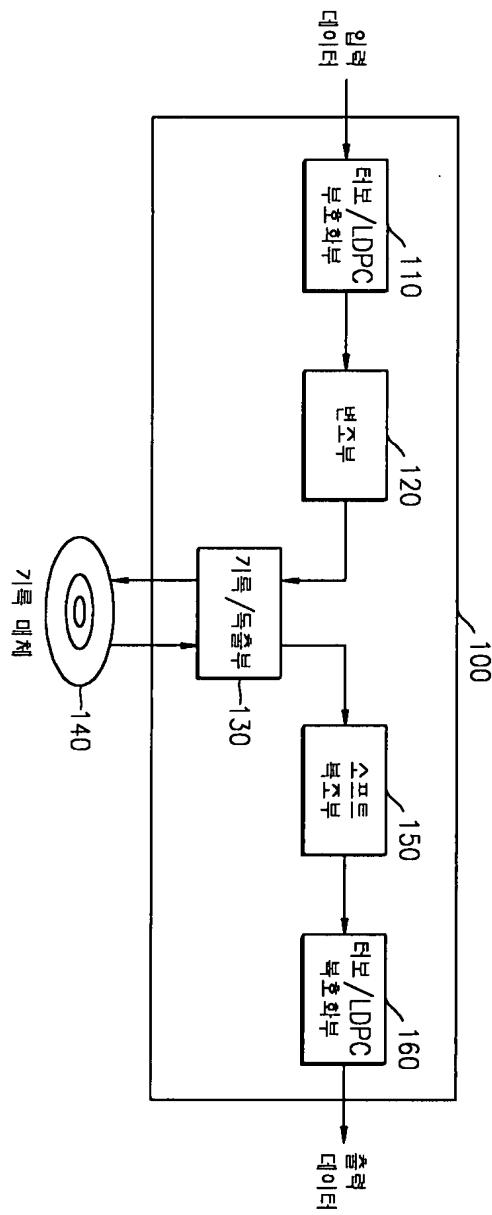
제7항에 있어서,

디코딩 테이블의 엔트리값 계산시, 상기 대응하는 부분합이 소정의 임계치를 초과하는지를 판단하는 임계치 판단부를 더 포함하고,

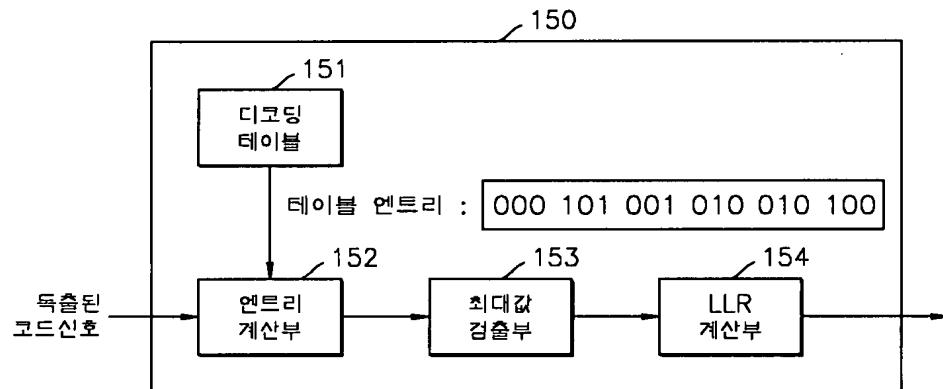
상기 임계치 판단부의 판단 결과, 상기 대응하는 부분합이 소정의 임계치를 초과하지 않는 경우에 상기 엔트리 계산부는 그 엔트리값은 계산하지 않는 것을 특징으로 하는 소프트 복조 장치.

## 【도면】

【도 1】



【도 2】



【도 3】

	$d_k d_{k+1}$
1	000 101 001 010 010 100
2	010 010 001 010 010 100
...	...
5000	000 101 001 010 010 010

【보 4】

수신된 코드번호 :

$r_0$	$r_1$	$r_2$	$r_3$	$r_4$	$r_5$	$r_6$	$r_7$	$r_8$	$r_9$	$r_{10}$	$r_{11}$	$r_{12}$	$r_{13}$	$r_{14}$	$r_{15}$	$r_{16}$	$r_{17}$
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	----------	----------	----------	----------	----------	----------	----------	----------

410

데이터 번트리 :

0	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

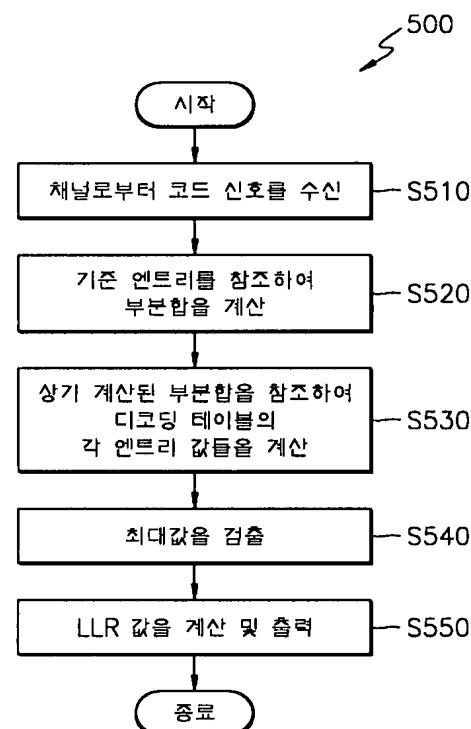
420

계산 결과 :

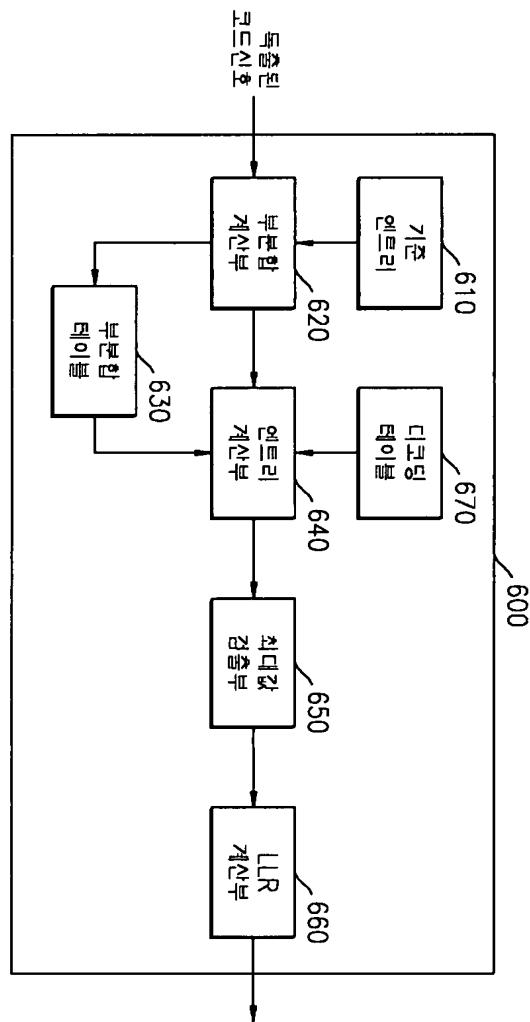
$-r_0 -r_1 -r_2 +r_3 -r_4 +r_5 -r_6 -r_7 +r_8 -r_9 +r_{10} -r_{11} -r_{12} +r_{13} -r_{14} +r_{15} -r_{16} -r_{17}$
---

430

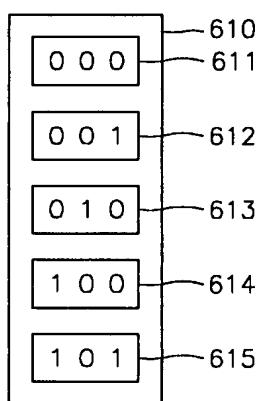
## 【도 5】



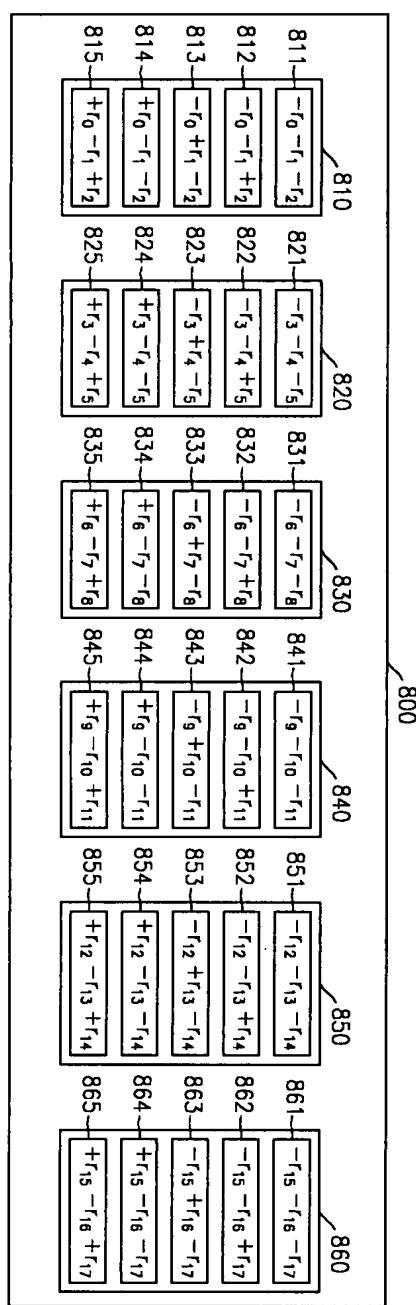
【도 6】



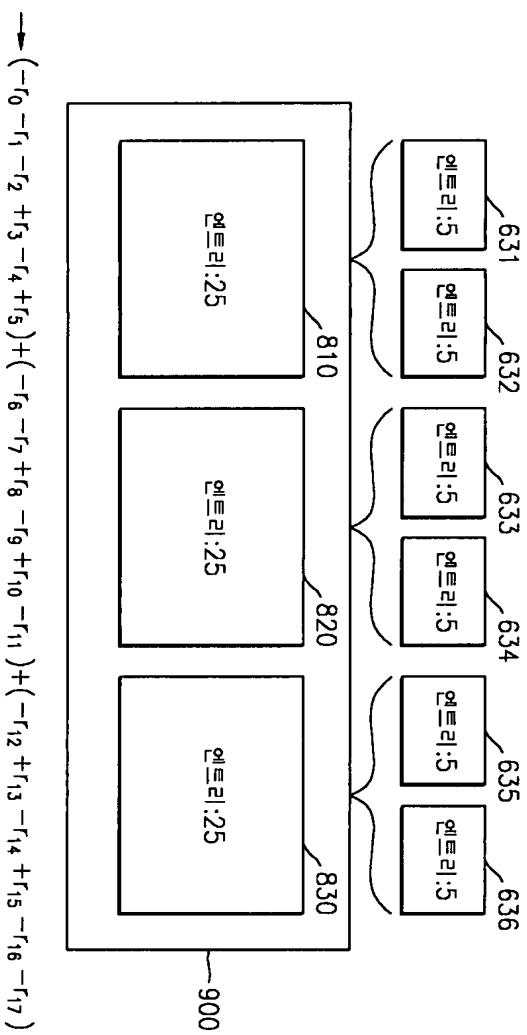
【도 7】



【H】 8]



【도 9】

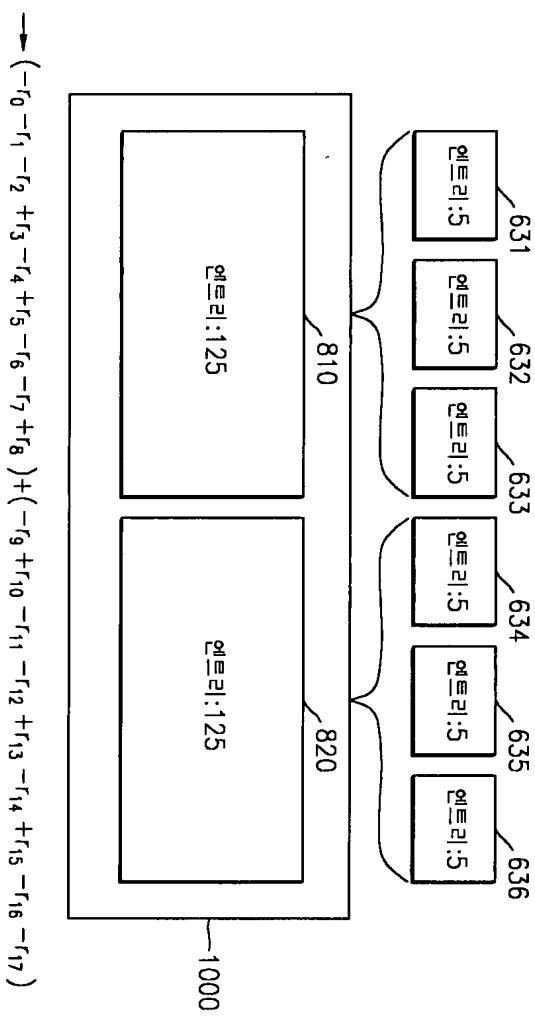


$$\rightarrow (-r_0 - r_1 - r_2 + r_3 - r_4 + r_5) + (-r_6 - r_7 + r_8 - r_9 + r_{10} - r_{11}) + (-r_{12} + r_{13} - r_{14} + r_{15} - r_{16} - r_{17})$$

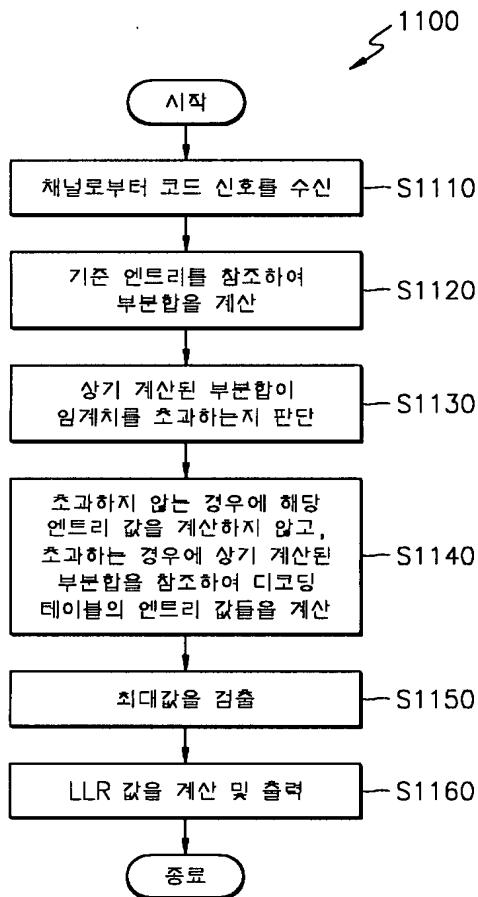
1020030005305

출력 일자: 2004/1/16

【도 10】



【도 11】



【도 12】

